"On the use of debug and test infrastructures for validating microprocessor based dependable systems through fault injection"

André V. Fidalgo[1,2]
[1]Instituto Superior de Engenharia do Porto
avf@dee.isep.ipp.pt

Gustavo R. Alves[1]
[1]Instituto Superior de Engenharia do Porto
galves@dee.isep.ipp.pt

José M. Ferreira[2]
[2]Faculdade de Engenharia da Universidade do Porto
jmf@fe.up.pt

Abstract

In recent years, there has been a rapid increase in the use of microprocessor-based systems in critical areas where failures imply risks to human lives, the environment or expensive equipment. One solution for avoiding a possible disaster lays in the use of dependable systems, able to tolerate and eventually correct faults, requiring high quality validation & verification in their development cycle. The PhD thesis here described aims to contribute a methodology that reuses a proposed standard debug & test infrastructure (NEXUS 5001) to access the microprocessor core with the objective of supporting the validation and verification steps of the fault-tolerant mechanisms through fault injection campaigns. For the purpose of later demonstrating the proposed methodology, a target microprocessor available as a synthesisable core for programmable components will be used. This aspect is crucial because it allows us to implement a prototype for demonstration purposes on a reconfigurable device. From these elements a fault injection infrastructure that can be utilized not only for validating the fault-tolerant characteristics of microprocessors but also for debugging and data collecting operations will be developed.

1. Introduction

The motivation behind the PhD work described in this document originates from the belief that it is possible and useful to develop a fault injection methodology through fault injection campaigns using a standard access infrastructure providing a great increase in performance and capabilities on the process of microprocessor design validation.

The proposed work will start with a study of the fault-tolerant capabilities of critical systems and the fault tolerance validation techniques currently used.

This study will provide the basis for the design of a system that implements a standard debug & test infrastructure on a target microprocessor both developed in VHDL format. These can then be used to implement a prototype on a reconfigurable device to validate the microprocessor fault tolerance capabilities through fault injection campaigns.

2. State of the Art

In safety critical computer based applications dependability is of utmost importance. Dependable systems are designed to detect errors that originate from software or hardware faults and recover from them maintaining acceptable operating conditions. The verification and validation of these systems is an important and hard to handle problem, although benefiting from some proposed solutions such as: analytical modeling, experimental techniques and fault injection [1]. Fault injection is recognized as a powerful solution, particularly to measure the effectiveness of the error detection mechanisms. It consists of injecting faults in the system components, while functional applications are being executed, and then observing the system response.

The hardest part of this approach is the methodology for actually injecting the fault, namely how to access those elements of the microprocessor where faults are more probable, generally the memory elements and communication buses.

The efficiency of a fault injection technique depends on the controllability and observability level of each microprocessor. Nowadays, almost every microprocessor comes with a debug & test infrastructure which provides a reasonable mean to access its core. However, such infrastructures are generally based on different architectures and access ports, normally requiring specific hardware and often with proprietary parts.

IEEE-ISTO 5001-1999, the NEXUS 5001 Forum Standard for a Global Embedded Processor Debug Interface [2] is an open industry standard that provides a general-purpose interface for the software development and debug of embedded processors. This proposed standard is an interesting possibility

for the development of a common fault injection methodology for the verification and validation of critical microprocessor-based systems

3. Implementation

The initial background study will be followed by a development and implementation process.

The first step will be the selection of a suitable microprocessor model and its implementation on a commercial programmable device. A possible target processor would be the LEON [3] which is available as a VHDL model and can be implemented in several commercial devices. This microprocessor characteristics and availability make it a viable target for demonstrating the implementation of the fault injection methodology that we propose to develop. Generic and fault tolerant applications will then be run on the target system and will be verified both in simulation and on the implemented device.

A set of verification results will be generated to be used as a comparison basis for the results obtained latter. A considerable part of this step will be the development of the fault model for the target processor and the fault injection campaigns that need to be executed.

The next step will be the implementation of a debug & test infrastructure compatible with the IEEE ISTO 5001 - NEXUS draft standard to be included in the target model. This infrastructure will also be modelled in VHDL and it will be tested first in simulation and later it will be migrated to the target processor model and synthesised with the complete system.

The implemented debug & test infrastructure will then be used to execute fault injection campaigns in the target microprocessor while this is running typical applications. Its functions will include the fault injection process and all the run-control and data-collecting operations used in fault injection campaigns.

The final step will be an analysis of the possibility of modifying the NEXUS debug & test infrastructure to provide extra functionalities. The proposed modifications will then be tested using the developed prototype to evaluate its added benefits.

4. Expected Results

The results obtained from the fault injection campaigns will allow an evaluation of the fault injection capabilities of the implemented solution.

Additionally a thorough analysis of the weight of the additional logic and its effects on system performance will also be performed.

These results should allow an adequate evaluation and quantification of the merits (and limitations) of the use of the NEXUS infrastructure for fault injection purposes.

Additionally the described work should provide a better view on the possibility of modifying (or extending) the NEXUS debug & test specification to better support fault injection tasks for the validation and verification of microprocessor systems.

The objective would be advanced fault injection features in a common and standardized infrastructure that can be used in a wide range of microprocessors.

Possible new features are added synchronization capabilities, new modes dedicated to fault injection operations and access to additional microprocessor components (like the processor registers or pipeline).

5. Related Work

The proposed work opens the way for several additional topics that may not be covered in the available time but will remain open for further study.

It may prove feasible to include the debug controller on the same programmable device where the target system is installed. In this manner the communication between controller and target may present considerable gains both in performance and robustness.

If the previous feature proves feasible then the next logical step would be to attempt to eliminate the host PC from the fault injection execution loop, by including all information and functionalities on a programmable device. This can be achieved with the inclusion of all the execution commands on onboard memory accessed by the embedded debug controller. This controller must be independent of the target microprocessor as to be able to run and communicate even if the target is stopped or non-responsive.

Some programmable devices provide to the user the possibility for on-the-fly reconfiguration. This feature can also be explored to test some interesting possibilities, for instance to make the target microprocessor configure the programmable device in which it is inserted so that it is possible to change the version of the debug & test infrastructure as dictated by the ongoing task.

References

[1] Ghani A. Kanawati et al, “FERRARI: A Flexible Software-Based Fault and Error Injection System”, IEEE transactions on computers 44, 1995.

[2] IEEE-ISTO 5001 [www.nexus5001.org], “The Nexus 5001 Forum Standard for a Global Embedded Processor Interface version 2.0”, 2003.

[3] Jiri Gaisler, “Concurrent error-detection and modular fault-tolerance in a 32-bit processing core for embedded space flight applications”, Fault-Tolerant Computing, 1994.